Embedding OPC Unified Architecture

Tom Hannelius, Mickey Shroff, Pasi Tuominen
Wapice Ltd, P.O.Box 8, FIN-65200 Vaasa, Finland
Tel: (06)-3194000, Fax: 06-3194001, email: firstname.lastname@wapice.com

KEYWORDS: OPC Unified Architecture, OPC UA, embedded device, interoperability, ANSI C

ABSTRACT

OPC UA is the newest in a series of specifications released by the OPC Foundation targeted to provide interoperability in industrial automation. OPC UA introduces fundamental architectural changes to earlier OPC specifications. These include e.g. a unified address space model, platform independence, security and scalability. Wapice Ltd has taken part in the OPC Unified Architecture (OPC UA) early adopter group (EAG) activities by contributing to the JAVA development of the OPC UA Specification.

This paper presents OPC UA basics with a focus on the OPC UA communication stack as well as scalability issues and OPC UA profiles. Embedded devices will utilize the ANSI C stack as a basis for transportation, security and encoding/decoding. Various ports of the stack have already been developed but the OPC Foundation is currently only providing a Windows port to its members. Porting the ANSI C stack to other platforms e.g. Linux or custom RTOS is essentially writing the platform dependent code on the portability layer of the stack. Porting to non Windows platforms is demonstrated by introducing the procedure of deploying OPC UA on Wapice's ARM9 based single board computer WARM.

1 INTRODUCTION

OPC has owned a position as the de facto standard within industrial automation system integration since the release of the OPC Data Access (DA) specification in the mid 90's. The OPC specifications address the ever growing need for software components of several manufactures to seamlessly communicate and understand each other by defining a standard interface and address space model.

OPC UA is the newest of the specifications and it introduces fundamental architectural changes to classical OPC concepts. While no more relying on Microsoft's COM/DCOM as the communication basis and rather utilizing industry standards such as XML Web Services and TCP/IP to name a few, OPC UA enables a much wider variety of application domains to adopt the technology. Now, everything from embedded platforms with limited resources to large scale enterprise systems independent of the programming language and the platform they are deployed on may be introduced to the new integration standard.

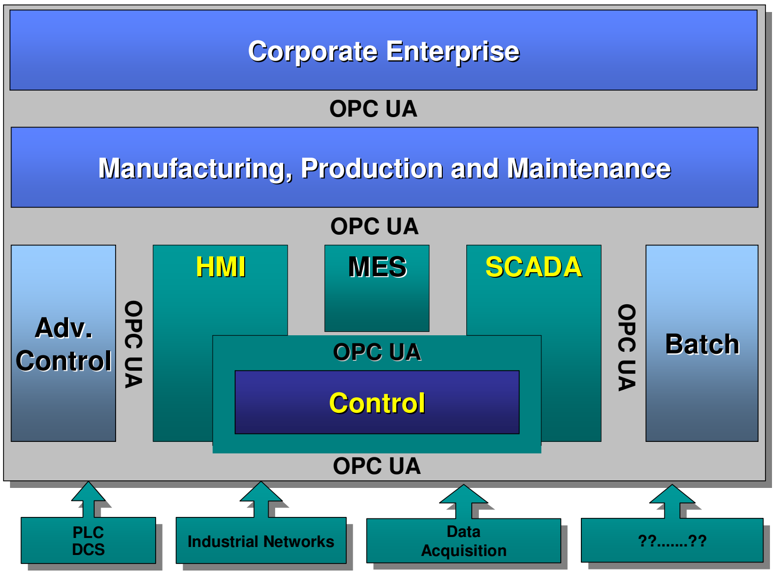

Figure 1 Scalable from embedded to enterprise /2/

By deploying the OPC UA address space model, applications may express real time data, alarms, history and events in the same address space. Formerly several separate specifications (Data Access, Historical Access, Alarms and Events, Commands) were to be utilized to achieve this. OPC UA is no longer based on Microsoft's COM/DCOM technology, but rather describes a service oriented architecture (SOA) for industrial applications relying on industry standards such as XML Web services. Security is built in to the base of OPC UA and the specification defines security mechanisms to be used on multiple levels in an OPC UA application. OPC UA may be deployed on applications from embedded devices to enterprise level systems. Scalability is provided through different profiles that define the services and parts of the OPC UA information model that an OPC UA application conforming to the respective profile must support as well as communication stack profiles (e.g. XML Web Services or UA Binary) they must provide /8/.

2 OPC UA CONCEPTS

An OPC UA application, being it a client or a server, builds upon some common components. Figure 2 presents building blocks of an OPC UA application.

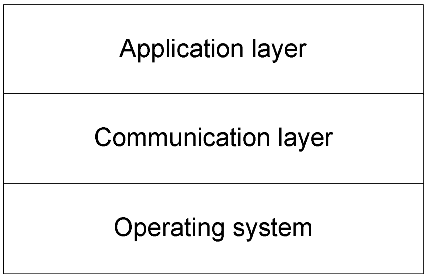

Figure 2 OPC UA building blocks

The bottom level illustrates the operating system (OS) that provides platform specific functionality to e.g. handle low level OS dependent transportation issues. The communication layer consists of the OPC UA communication stack. The OPC UA communication stack is an essential component in any OPC UA application and handles fundamental tasks common to all OPC UA clients and servers. These include the serialization of OPC UA request and response messages, securing the serialized byte stream to guard against security threats such as message alteration and eavesdropping, and transmitting the serialized and secured message to/from the OPC UA counterpart on the other end of the transmission medium. The application layer contains the implementation of OPC UA services defined in part 4 of the OPC UA Specification /4/.

2.1 The OPC UA communication stack

The communication stack represents OPC UA's interoperability layer and provides means to map the abstract set of protocol independent services /4/ into the underlying protocol, depending on what the protocol is. The OPC UA specification currently defines the communication stack profiles illustrated in Figure 3. The OPC UA communication stack architecture does not constrain possible additional mappings to be deployed on the stack. Rather, it is expected that other protocols will be added in the future.

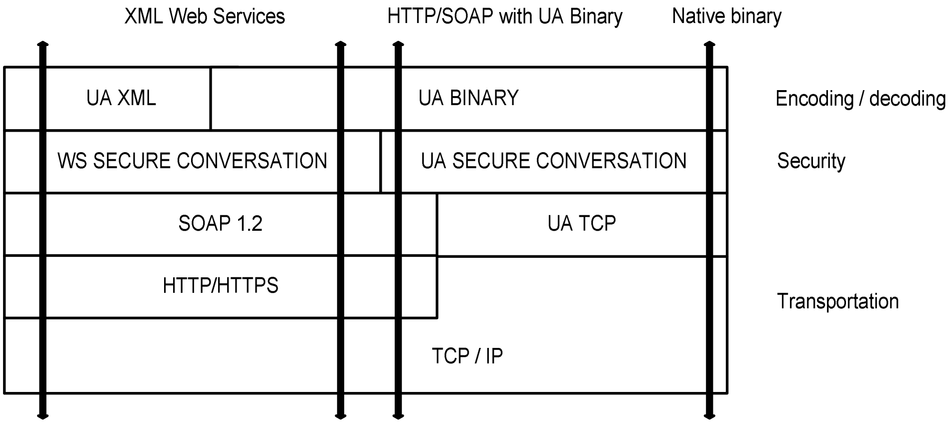

Figure 3 OPC UA communication stack technology mappings /7/

The serialization mechanisms i.e. UA XML and UA Binary defined in part 6 of the OPC UA specification /5/ are illustrated on top. Both provide a mapping for the representation of OPC UA request and response messages on the communication medium. UA XML provides an XML based encoding schema targeted for use with Web Services and UA Binary a proprietary UA binary schema optimized for high encoding/decoding speed.

The different security mechanisms are illustrated on the second level. When securing messages with WS Secure Conversation, the communication stack relies on standard protocols defined by Web Services Security (WS-Security). UA Secure Conversation on the other hand is a binary equivalent of the WS Secure Conversation algorithms aimed for high performance communication stacks /3/.

The transport mechanisms are illustrated from the third level down with SOAP/HTTP (Simple Object Access Protocol above Hypertext Transfer Protocol) for Web Services and UA TCP for high transportation performance needs. The OS dependent TCP layer forms the basis for all transportation mechanisms.

While the XML Web Services and HTTP/SOAP with UA Binary communication stack profiles provide e.g. optimal tool support and are firewall friendly, the Native Binary profile provides the best performance and is the communication stack profile that will be utilized in performance and resource sensitive environments e.g. embedded devices.

3 SCALABLE FROM EMBEDDED TO ENTERPRISE LEVEL

The OPC UA specification part 7 /6/ defines *Profiles* to address the fact that resource availability and consumption differ drastically from embedded boards to high end enterprise mainframes and all between. A profile describes the capabilities of an OPC UA application in terms of e.g. the services, communication stack profile and parts of the OPC UA information model the application supports. Profiles enable OPC UA clients and servers to describe the features they provide in a standard way and thus relieves the need for implementing custom functionality to query the capabilities of OPC UA counterparts. There are several profiles currently defined in the OPC UA specification and it is expected that new profiles will be added in the future.

3.1 Embedded profiles

The OPC UA server profiles targeted for embedded devices include *Low End Device Server, Embedded UA Server* and *Standard UA Server* /6/.

The Low End Device Server profile defines a bare bone functionality set that low end devices with limited resources may comply to. The core functionality of any OPC UA server such as address space and session management as well as attribute services e.g. Read is included.

The Embedded UA Server profile defines e.g. subscription services, in addition to the Low End Device Server functionalities, and is aimed for devices with more than 50 megabytes of memory and with CPUs comparable to an Intel 486 processor.

The Standard UA Server profile defines most of the common functionality that OPC UA servers shall support including discovery services in addition to the aforementioned profile’s functionality /6/.

4 PORTING THE ANSI C STACK

The OPC UA ANSI C communication stack is a platform independent component that can be compiled on any platform that provides an ANSI C compiler /1/. However, the current version of the stack provided to OPC Foundation members contains only a Windows port and thus deploying it in any other environment requires porting. Figure 4 illustrates the ANSI C communication stack layout.

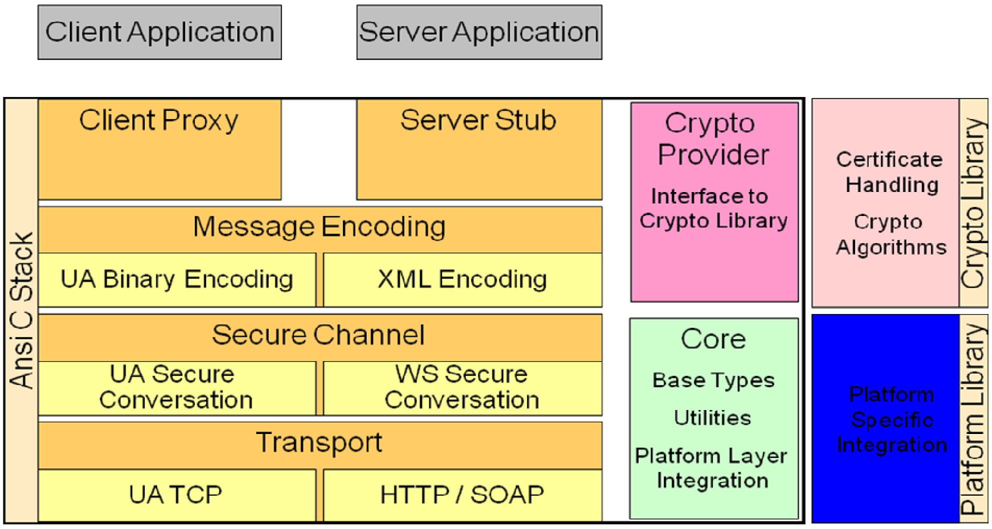

Figure 4 ANSI C stack layout /7 /

Porting the ANSI C communication stack is essentially writing the platform dependent code on the portability layer of the stack (Platform Specific Integration block in Figure 4). The ported functionality includes utilizing platform specific primitives for e.g. socket handling, synchronization and cryptography. Porting the ANSI C stack to an embedded board was demonstrated by Wapice Ltd by deploying an OPC UA server application on our ARM9 based single board computer WARM. As a result WARM is equipped with OPC UA communication capability conforming to the Standard UA Server profile.

5 CONCLUSIONS

OPC UA will evidently be the next generation integration standard – a continuum to the classical, widely used OPC specifications. OPC UA addresses deficiencies of the classical OPC specifications by defining e.g. a uniform address space model, support for custom information models and platform independence. The new features will enable a much larger field of application domains to embrace the technology and it is expected that OPC UA will be seen on everything from embedded devices to large scale enterprise systems.

Wapice Ltd has been a member of the OPC Foundation since 2001 and an active participant in the OPC UA Early Adopter Group contributing to the Java development of the OPC UA specification. Wapice Ltd has demonstrated the porting of the OPC UA ANSI C communication stack to a non Windows environment and developed an OPC UA server application on our ARM9 based single board computer WARM.

6 REFERENCES

1. Matthias Damm: OPC Unified Architecture, OPC Goes Embedded, OPC Day Finland, October 2008

2. OPC Foundation: OPC Unified Architecture Specification Part 1: Overview and concepts, Release 1.01, February 2009

3. OPC Foundation: OPC Unified Architecture Specification Part 2: Security Model, Release 1.01, February 2009

4. OPC Foundation: OPC Unified Architecture Specification Part 4: Services, Release 1.01; February 2009

5. OPC Foundation: OPC Unified Architecture Specification Part 6: Mappings, Release 1.00; February 2009

6. OPC Foundation: OPC Unified Architecture Specification Part 7: Profiles, Release 1.00, February 2009

7. OPC Foundation: DevCon 2007, OPC Unified Architecture, Conference Material, 2007

8. Tom Hannelius, Mikko Salmenperä, Seppo Kuikka: Roadmap to adopting OPC UA, IEEE INDIN, Daejeon, Korea, July 2008